

Amendments to the Claims

Please amend Claims 1, 15, 21, 33, cancel Claim 31, and add new Claim 35, all as shown below. Applicant respectfully reserves the right to prosecute any originally presented or canceled claims in a continuing or future application.

1. (Currently Amended) A system for supporting application deployment, comprising:
 - a plurality of deployment descriptor files that are adapted to describe deployment and configuration information of a plurality of applications deployed on a web server, wherein each application of the plurality of applications is associated with at least one deployment descriptor file of the plurality of deployment descriptor files and each ~~[[said]]~~ application is deployed in a project directory of ~~a plurality of directories~~ in the web server; and
 - a builder component capable of
 - creating a ~~master first~~ first tree data structure based on ~~a present state of all the plurality of~~ deployment descriptor files, wherein the ~~master first~~ first tree data structure represents a state of resources associated with the plurality of applications that are described in the plurality of deployment descriptor files, wherein the ~~master first~~ first tree data structure includes a first sub-tree that represents the state of resources that corresponds to a first application of the plurality of applications;
 - creating a ~~first separate second~~ first separate second tree data structure, independent of the plurality of deployment descriptor files, based on a current state of source files existing in a first project directory of ~~the plurality of directories~~ associated with the first application, wherein the ~~first separate second~~ first separate second tree data structure represents the current state of resources associated with the first application;
 - comparing the first sub-tree that corresponds to the first application in the ~~master first~~ first tree data structure with the ~~first separate second~~ first separate second tree data structure; and
 - refreshing the ~~master first~~ first tree data structure based on the ~~first separate second~~ first separate second tree data structure if the first sub-tree in the ~~master first~~ first tree data structure is different from the ~~first separate second~~ first separate second tree data structure.
2. (Previously Presented) The system of claim 1, further comprising:
 - a user interface capable of rendering an error message.

3. (Previously Presented) The system of claim 2 wherein:
a user selection of the error message can cause a second user interface to render a user-editable representation of the at least one deployment descriptor file that is in error.
4. (Previously Presented) The system of claim 1, further comprising:
a parser capable of generating a representation of the at least one deployment descriptor file;
a generator capable of creating the at least one deployment descriptor file; and
a validator capable of validating the at least one deployment descriptor file.
5. (Previously Presented) The system of claim 4 wherein:
the validator is capable of generating an error when it encounters a syntactic or semantic fault in the at least one deployment descriptor file.
6. (Previously Presented) The system of claim 1, wherein:
the builder component is further capable of automatically updating the at least one deployment descriptor file to reflect one or more changes in at least one source code file associated with an application in the plurality of applications.
7. (Previously Presented) The system of claim 1, further comprising:
a hierarchical representation that can include information pertaining to an archive file.
8. (Previously Presented) The system of claim 1, wherein:
the at least one deployment descriptor file can be expressed as an Extensible Markup Language document.
- 9-14. (Canceled).
15. (Currently Amended) A method for supporting application deployment, comprising:
deploying a plurality of applications on a web server, wherein each application of the plurality of applications is associated with at least one deployment descriptor file of a plurality of deployment descriptor files that describes deployment and configuration information of the

plurality of applications on the web server, and each ~~[[said]]~~ application is deployed in a project directory of ~~a plurality of directories~~ in the web server;

creating a master first tree data structure based on ~~a present state of all~~ the plurality of deployment descriptor files, wherein the ~~master first~~ tree data structure represents a state of resources associated with the plurality of applications that are described in the plurality of deployment descriptor files, wherein the ~~master first~~ tree data structure includes a first sub-tree that represents the state of resources that corresponds to a first application of the plurality of applications;

creating a ~~first separate~~ second tree data structure, independent of the plurality of deployment descriptor files, based on a current state of source files existing in a first project directory of ~~the plurality of directories~~ associated with the first application, wherein the ~~first separate~~ second tree data structure represents the current state of resources associated with the first application;

comparing the first sub-tree that corresponds to the first application in the ~~master first~~ tree data structure with the ~~first separate~~ second tree data structure; and

refreshing the master first tree data structure based on the ~~first separate~~ second tree data structure if the first sub-tree in the ~~master first~~ tree data structure is different from the ~~first separate~~ second tree data structure.

16. (Previously Presented) The method of claim 15, further comprising:

providing a parser capable of generating a representation of the at least one deployment descriptor file;

providing a generator capable of creating the at least one deployment descriptor file; and

providing a validator capable of validating the at least one deployment descriptor file.

17. (Previously Presented) The method of claim 16, further comprising:

generating, via the validator, an error when it encounters a syntactic or semantic fault in the at least one deployment descriptor file.

18. (Previously Presented) The method of claim 15, further comprising:

providing a builder component capable of automatically updating the at least one deployment descriptor file to reflect one or more changes in at least one source code file associated with an application in the plurality of applications.

19. (Previously Presented) The method of claim 15 wherein:
including a hierarchical representation information pertaining to an archive file.
20. (Previously Presented) The method of claim 15 wherein:
expressing the at least one deployment descriptor file as an Extensible Markup Language document.
21. (Currently Amended) A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:
- deploy a plurality of applications on a web server, wherein each application of the plurality of applications is associated with at least one deployment descriptor file of a plurality of deployment descriptor files that describes deployment and configuration information of the plurality of applications on the web server, and each said application is deployed in a project directory of a plurality of directories in the web server;
- create a master first tree data structure based on ~~a present state of all the plurality of~~ deployment descriptor files, wherein the master first tree data structure represents a state of resources associated with the plurality of applications that are described in the plurality of deployment descriptor files, wherein the master first tree data structure includes a first sub-tree that represents the state of resources that corresponds to a first application of the plurality of applications;
- create a ~~first separate second~~ second tree data structure, ~~independent of the plurality of deployment descriptor files~~, based on a current state of source files existing in a first project directory of the plurality of directories associated with the first application, wherein the ~~first separate second~~ second tree data structure represents the current state of resources associated with the first application;
- compare the first sub-tree that corresponds to the first application in the master first tree data structure with the ~~first separate second~~ second tree data structure; and
- refresh the master first tree data structure based on the ~~first separate second~~ second tree data structure if the first sub-tree in the master first tree data structure is different from the ~~first separate second~~ second tree data structure.
- 22.-27. (Canceled).

28. (Previously Presented) The system of claim 1, wherein:
the interactive tool is capable of automatically repairing a first deployment descriptor file
if the first deployment descriptor file is defective.
29. (Previously Presented) The system of claim 1, wherein:
the builder component is further capable of creating a tree data structure that embodies
hierarchical relationships of nested XML statements.
- 30.-31. (Canceled).
32. (Previously Presented) The system of claim 1, further comprising:
a first user interface capable of rendering a hierarchical representation of the
plurality of deployment descriptor files, wherein a component of the representation can
be selected by a user; and
a second user interface capable of rendering a user-editable representation of
the selected component.
33. (Currently Amended) The system of claim 1, wherein:
the builder component is further capable of generating a new deployment
descriptor file for the first application from the refreshed ~~master~~ the first tree data
structure.
34. (Previously Presented) The system of claim 1, further comprising:
a pane that displays a single field for a value, wherein the single field maps to
multiple values in the at least one deployment descriptor file.
35. (New) The system of claim 1, wherein:
the builder component is further capable of
detecting that a module exists in the first tree data structure and the
module does not exist in the second tree data structure; and

preventing deleting the module from the first tree data structure, when another application depends on the at least one deployment descriptor file associated with the first application.